IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re patent application of:
Bhide et al.

Atty. Docket No.: JP920030164US1

Serial No.: 10/729,166

Group Art Unit: 2165

Filed: December 5, 2003

Examiner: Syed, Farhan M.

For: OPTIMAL USE OF TRIGGERS FOR DETECTING DATABASE EVENTS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' APPEAL BRIEF

Sirs:

Appellants respectfully appeal the final rejection of claims 1-20 in the Office Action dated October 6, 2006. A Notice of Appeal was timely filed on January 8, 2007.

I. REAL PARTY IN INTEREST

The real party in interest is International Business Machines Corporation, Armonk, New York, assignee of 100% interest of the above-referenced patent application.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 1-20, all the claims pending in the application and set forth fully in the attached appendix (Section IX), are under appeal. Claims 1-15 were originally filed in the application on December 5, 2003 contemporaneous with a Preliminary Amendment. A non-final Office Action was issued on May 19, 2006 rejecting claims 1-15. The Appellants filed an Amendent under 37 C.F.R. §1.111 on July 28, 2006 amending claims 1-6 and 10-15 and adding claims 16-20. A final Office Action was issued on October 6, 2006 rejecting claims 1-20. The Appellants filed a Response under 37 C.F.R. §1.116 on December 4, 2006. An Advisory Action was issued on January 5, 2007 indicating that the Response filed under 37 C.F.R. §1.116 on December 4, 2006 did not place the application in condition for allowance. The Appellants timely filed a Notice of Appeal on January 8, 2007.

Claims 1-20 stand rejected under 35 U.S.C. §102(b) as being anticipated by Dayal et al. (“Active Database Systems,” Sept. 1994), hereinafter referred to as “Dayal”. It should be noted that page 5 of the Office Action of October 6, 2006 gives the date of the Dayal publication as

September 2004. It is understood by Appellants that this is a typographical error in the Office Action, and that the correct publication date of Dayal is September 1994. Appellants respectfully traverse these rejections based on the following discussion.

IV. STATEMENT OF AMENDMENTS

A final Office Action dated October 6, 2006 stated all the pending claims 1-20 were rejected. The Appellants filed a Response under 37 C.F.R. §1.116 on December 4, 2006. An Advisory Action was issued on January 5, 2007 indicating that the Response filed under 37 C.F.R. §1.116 on December 4, 2006 did not place the application in condition for allowance. The claims shown in the appendix (Section IX) are shown in their amended form as of the July 28, 2006 Amendment (and December 4, 2006 Response).

V. SUMMARY OF CLAIMED SUBJECT MATTER

The Appellants' claimed invention is generally described in pages 3 through 21 of the specification and shown in Figures 1 through 10 of the application. More specifically, with respect to the claimed subject matter (with the pages and line numbers of the Appellants' specification and figure numbers, in any, of the Appellants' application given in brackets below):

Claim 1: A method of monitoring events in a database [pp. 7, lines 14-15; database system 200 of FIG. 2], said method comprising: storing in said database [pp. 7, lines 14-15; database system 200 of FIG. 2] at least one database rule [pp. 7, lines 22-26; step 310 of FIG. 3]; mapping temporal constraints of an event of the database rule to corresponding temporal events [pp. 7, lines 23-24; pp. 8, lines 10-13; step 320 of FIG. 3]; changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule [pp. 7, lines 24-25; step 330 of FIG. 3]; registering alarms associated with a start and end of a lifespan of each temporal event [pp. 10, lines 22-26; temporal daemon 416 of FIG. 4]; selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints [pp. 7, lines 22-26; step 340 of FIG. 3]; and upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event [pp. 11, lines 16-23; dynamic trigger removal 417 of FIG. 4].

Claim 2: Removing from the database temporal events that cannot evaluate as true [pp. 11, lines 7-14; dynamic trigger removal 417 of FIG. 4].

Claim 3: Limiting the lifespan of an event to the overlapping period of the lifespan of a parent event [pp. 10, lines 8-15; page 12, lines 4-12; page 15, lines 7-16 of FIG. 4].

Claim 4: Changing the lifespan of an event to omit periods in which the event cannot evaluate as true [pp. 8, lines 15-16; pp. 12, lines 11-33].

Claim 5: Assigning a lifespan of an event having an undefined lifespan as the lifespan of a parent event [pp. 15, lines 1-5; steps 540 and 545 of FIG. 5A].

Claim 6: Propagating the lifespan or context of the parent node to all children nodes of the parent node [pp. 14, lines 20-26; step 520 of FIG. 5A].

Claim 7: A lifespan of an event is expressed as a predetermined duration of time [pp. 6, lines 18-23].

Claim 8: The lifespan is dependent upon the associated event [pp. 5, line 27 through pp. 6, line 6].

Claim 9: The lifespan ends at a predetermined time, or recurs at a predetermined period of time [pp. 5, line 27 through pp. 6, line 6].

Claim 10: Combining events using a sequence operator to form a composite event having a time span [pp. 6, lines 8-15].

Claim 11: Associating a lifespan with the sequence operator [pp. 5, lines 16-25 (as amended in July 28, 2006 Amendment)].

Claim 12: Storing a database rule as an event-condition-action (ECA) rule [pp. 6, lines 25-28].

Claim 13: A database [pp. 7, lines 14-15; database system 200 of FIG. 2] recorded on a computer storage medium [pp. 20, lines 7-11; computer system 1000 of FIG. 10] comprising: software code means for [programming language; pp. 19, lines 4-27; pp. 20, lines 7-11; pp. 21, lines 5-8] storing in said database at least one database rule [pp. 7, lines 22-26; step 310 of FIG.

3]; software code means for [programming language; pp. 19, lines 4-27; pp. 20, lines 7-11; pp. 21, lines 5-8] mapping temporal constraints of an event of the database rule to corresponding temporal events [pp. 7, lines 23-24; pp. 8, lines 10-13; step 320 of FIG. 3]; software code means for [programming language; pp. 19, lines 4-27; pp. 20, lines 7-11; pp. 21, lines 5-8] changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule [pp. 7, lines 24-25; step 330 of FIG. 3]; software code means for [programming language; pp. 19, lines 4-27; pp. 20, lines 7-11; pp. 21, lines 5-8] registering alarms associated with a start and end of a lifespan of each temporal event [pp. 10, lines 22-26; temporal daemon 416 of FIG. 4]; software code means for [programming language; pp. 19, lines 4-27; pp. 20, lines 7-11; pp. 21, lines 5-8] selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints [pp. 7, lines 22-26; step 340 of FIG. 3]; and software code means for [programming language; pp. 19, lines 4-27; pp. 20, lines 7-11; pp. 21, lines 5-8], upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event [pp. 11, lines 16-23; dynamic trigger removal 417 of FIG. 4].

Claim 14: A system [pp. 9, lines 15-25 (as amended in July 28, 2006 Amendment); architectural framework 400 of FIG. 4] for monitoring events in a database [pp. 7, lines 14-15; database system 200 of FIG. 2], said system [pp. 9, lines 15-25 (as amended in July 28, 2006 Amendment); architectural framework 400 of FIG. 4] comprising: means for [storage device 1055 of FIG. 10] storing in said database at least one database rule [pp. 7, lines 22-26; pp. 20, lines 25-26; step 310 of FIG. 3]; means for [processor 1040 of FIG. 10; pp. 20, line 14] mapping temporal constraints of an event of the database rule to corresponding temporal events [pp. 7, lines 23-24; pp. 8, lines 10-13; step 320 of FIG. 3]; means for [temporal condition identifier 205 of FIG. 2; pp. 7, lines 15-16] changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule [pp. 7, lines 24-25; step 330 of FIG. 3]; means for [temporal daemon 416 of FIG. 4] registering alarms associated with a start and end of a lifespan of each temporal event [pp. 10, lines 22-26;]; means for selectively deploying [dynamic trigger deployment 418 of FIG. 4, pp. 10, line 28 through pp. 11, line 5 (as amended in July 28, 2006 Amendment)] and selectively permanently removing [dynamic trigger removal 417 of FIG. 4; pp. 11, lines 7-23] the temporal events from said database based upon the changed temporal constraints [pp. 7, lines 22-26; step 340 of FIG. 3]; and means for [dynamic trigger removal 417 of FIG. 4], upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event [pp. 11, lines 16-23].

Claim 15: A program storage device [pp. 20, lines 7-11; computer system 1000 of FIG. 10] readable by computer [pp. 20, line 13; computer 1020 of FIG. 10], tangibly embodying a program of instructions [pp. 20, lines 2-11] executable by said computer [pp. 20, line 13; computer 1020 of FIG. 10] to perform a method of monitoring events in a database [pp. 7, lines 14-15; database system 200 of FIG. 2], said method comprising: storing in said database at least one database rule [pp. 7, lines 22-26; step 310 of FIG. 3]; mapping temporal constraints of an event of the database rule to corresponding temporal events [pp. 7, lines 23-24; pp. 8, lines 10-13; step 320 of FIG. 3]; changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule [pp. 7, lines 24-25; step

330 of FIG. 3]; registering alarms associated with a start and end of a lifespan of each temporal event [pp. 10, lines 22-26; temporal daemon 416 of FIG. 4]; selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints [pp. 7, lines 22-26; step 340 of FIG. 3]; and upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event [pp. 11, lines 16-23; dynamic trigger removal 417 of FIG. 4].

Claim 16: Using a separate device external to said database to detect the combined events [pp. 9, lines 1-8; pp. 11, lines 3-5; event monitor 425 of FIG. 4].

Claim17: Said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database [pp. 3, lines 5-7].

Claim 18: Combining events using a sequence operator to form a composite event having a time span [pp. 6, lines 8-15].

Claim 19: Using a separate device external to said database to detect the combined events [pp. 9, lines 1-8; pp. 11, lines 3-5; event monitor 425 of FIG. 4].

Claim 20: Said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database [pp. 3, lines 5-7].

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The issues presented for review by the Board of Patents Appeals and Interferences are whether claims 1-20 are unpatentable under 35 U.S.C. §102(b) as being anticipated by Dayal.

VII. ARGUMENT

A. The Prior Art Rejections Based of Claims 1-20

1. The Position in the Office Action

As per claims 1 and 13-15, the Office Action of October 6, 2006 states that Dayal teaches a method of monitoring events in a database, said method comprising: (i.e. "An active database system, in contrast, is a database system that monitors situations of interest, and when they occur, triggers an appropriate response in a timely manner." According to the Office Action, the preceding text clearly indicates that monitoring events in a database is an active database system.) (Page 1, paragraph 3); storing in said database at least one database rule (i.e. "The desired behavior is expressed in production rules (also called event-condition-action rules), which are defined and stored in the database." According to the Office Action, the preceding text clearly indicates that storing production rules indicates that at least one rule is stored in the database.) (Page 1, paragraph 3); mapping temporal constraints of an event of the database rule to corresponding temporal events (i.e. "An inference engine cycles through all the rules in the system, matching the condition parts of the rules with the data in working memory." According to the Office Action, the preceding text clearly indicates that mapping is matching and temporal constraints are conditions that contain time and event.) (Page 1, paragraph 3); changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of the database rule (i.e. "The action part may modify the working memory, possibly according to the matched data, and the cycle continues until no more rules match." According to the Office Action, the preceding text clearly indicates that changing the temporal constraints is modifying the condition, which is the action part of the working memory.) (Page 1, paragraph 3); registering alarms associated with a start and end of a lifespan of each temporal event (i.e. "...a rule is triggered whenever one or more of its triggering operation occurs." "In addition, active database systems must provide mechanisms for event detection and rule triggering, for condition testing, for rule action execution, and for user development of rule applications." According to the Office Action, the preceding text clearly indicates that registering alarms associated with each temporal event is an illustration of a triggering operation. Furthermore, according to the Office Action, using an active database, an ordinary person skilled in the art may reasonably infer that such alarms are anticipated in the database system, because for an active database system to be active, it must do event detection, and registering alarms illustrates such a point.)(Page 14, section 3.3; Pages 17-18; section 4); selectively deploying and selectively permanently removing the temporal events from database based upon said changed temporal constraints (i.e. "Rules are structured objects, having events, conditions and actions as their components. According to the Office Action, like any object, rules can be created, deleted, or modified. In addition, according to the Office Action, rule objects have some special operations, including: fire, which causes a rule to be triggered; enable, which cases a rule to be activated; disable, which causes a rule to be deactivated (so that it won't be triggered even if its triggered event occurs." According to the Office Action, "[r]ules refer to particular tables, and so are subject to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative." According to the Office Action, the preceding text clearly indicates that selectively deploying is enable, which cases the trigger to be activated and selectively removing is disable, which cases the trigger to be deactivated. Furthermore when a table is dropped, according to the Office Action, an ordinary person skilled in the art anticipates that the table is either deleted or removed from the database, thus it can be inferred that the table is permanently removed; which by proxy would infer the temporal events would too be permanently removed.) (Page 3, paragraph 3; Page 9, section 2.5); upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarms associated with the permanent removed temporal event (i.e. "Rules refer to particular tables, and so are subject to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative." When a table is dropped, according to the Office Action, an ordinary person skilled in the art may reasonably infer that the skilled person can remove the table from the database and thereby manually remove all temporal constraints and alarms associated with the targeted table. The manual removal of the temporal constraints and associated alarms are consistent with a natural human phenomenon, when the skilled person performs duly maintenance on the database, according to the Office Action. Whether manually or dynamically performed, no novelty, according to the Office Action, exists in optimizing system efficiency.) (Page 9, section 2.5).

As per claim 2, the Office Action indicates that Dayal teaches a method further comprising removing from the database temporal events that cannot evaluate as true (i.e. "Most database rule systems handle errors during rule processing by aborting the current transaction, since this is how conventional database systems typically handle errors during transaction processing.") (Page 17, paragraph 2).

As per claim 3, the Office Action indicates that Dayal teaches a method further comprising limiting the lifespan of an event to the overlapping period of the lifespan of a parent event (i.e. "The nested transaction model used in HiPAC [High Performance ACtive Database System] allows some of these possibilities. When a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, according to the Office Action, failure can be propagated up the transaction tree all the way to the root (top) transaction.") (Page 17, paragraph 3).

As per claim 4, the Office Action indicates that Dayal teaches a method further comprising changing the lifespan of an event to omit periods in which the event cannot evaluate as true (i.e. "In general, when a triggered rule is executed in Ariel, the rule processes the entire set of triggering changes, including both the user-generated changes that initiated the rule processing and any subsequent changes made by rule actions." "Some languages have sequential execution semantics, while others allow concurrent execution. With either sequential or concurrent execution semantics, according to the Office Action, there is also the issue of whether one rule can trigger the execution of another rule or of (another instance of) the same rule. Clearly, according to the Office Action, if such nested triggering is allowed, termination is a concern. According to the Office Action, the preceding text clearly indicates that when the author mentioned termination is a concern, an ordinary person skilled in the art understands that termination or abort must take place when the event cannot evaluate as true.) (Page 12, paragraph 1; page 11, paragraph 1).

As per claim 5, the Office Action indicates that Dayal teaches a method further comprising assigning a lifespan of an event having an undefined lifespan as the lifespan of a parent event (i.e. 'in addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action." According to the Office Action, "[t]he nested transaction model used in HiPAC allows some of these possibilities. When a rule execution sub transaction fails, according to the Office Action, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, according to the Office Action, failure can be propagated up the transaction tree all the way to the root (top) transaction.") (Page 3, paragraph 5; page 17, paragraph 3).

As per claim 6, the Office Action indicates that Dayal teaches a method further comprising propagating the lifespan or context of the parent node to all children nodes of the parent node (i.e. "The nested transaction model used in HiPAC allows some of these possibilities. According to the Office Action, when a rule execution sub transaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, according to the Office Action, failure can be propagated up the transaction tree all the way to the root (top) transaction.") (Page 17, paragraph 3).

As per claim 7, the Office Action indicates that Dayal teaches a method wherein a lifespan of an event is expressed as a predetermined duration of time (i.e. "In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action." According to the Office Action, "[s]ome languages support rules triggered by temporal events. These might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday).") (Page 3, paragraph 5; page 5, paragraph 2) according to the Office Action.

As per claim 8, the Office Action indicates that Dayal teaches a method wherein the lifespan is dependent upon the associated event (i.e. "In addition, some languages provide mechanisms whereby data (parameters) can be bound in the event and/or condition part of a rule, then passed to the condition and/or action.") (Page 3, paragraph 5).

As per claim 9, the Office Action indicates that Dayal teaches a method wherein the lifespan ends at a predetermined time, or recurs at a predetermined period of time (i.e. "Some languages support rules triggered by temporal events. According to the Office Action, these might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday).") (Page 5, paragraph 2).

As per claims 10 and 18, the Office Action indicates that Dayal teaches a method further comprising combining events using a sequence operator to form a composite event having a time span (i.e. "Some languages support rules triggered by temporal events. According to the Office Action, these might be absolute (e.g.: 08:00:00 hours on January 1, 1994), relative (e.g., 5 seconds after takeoff), or periodic (e.g., 17:00:00 hours every Friday)." According to the Office Action, "[w]hen an instance of this event type occurs, the formal parameters are bound to a specific employee (the one whose salary is being updated) and two specific integers (this employee's old salary and new salary).") (Page 5, paragraph 2; page 6, paragraph 3).

As per claim 11, the Office Action indicates that Dayal teaches a method further comprising associating a lifespan with the sequence operator (i.e. "Most importantly, unlike in AI systems, in active database systems rule processing is integrated with conventional database activity -queries, modifications, and transactions- and it is this activity that causes rules to become triggered and initiates rule processing." According to the Office Action, the preceding text clearly indicates that a lifespan is contained within a rule that processes queries, modifications, and transactions, and the sequence operator is the active database system that initiates rule processing.) (Page 10, paragraph 3).

As per claim 12, the Office Action indicates that Dayal teaches a method further comprising the step of storing a database rule as an event-condition-action (ECA) rule (i.e. "The desired behavior is expressed in production rules (also called event-condition-action rules), which are defined and stored in the database." According to the Office Action, the preceding text clearly indicates that ECA rules are stored in a database.) (Page 1, paragraph 3).

As per claims 16 and 19, the Office Action indicates that Dayal teaches a method further comprises using a separate device external to said database to detect the combined events (i.e. "The implementation of an active database system can include many useful features that support the rule programmer. According to the Office Action, features for analyzing rule processing include the ability to trace rule execution, to display the current set of triggered rules, to query and browse the set of rules, and to cross-reference rules and data. According to the Office Action, other useful features include the ability to control errors in rule programs, to activate and deactivate selected rules or groups of rules while the database system is processing transactions, and to experiment with rules on an off-line subset of a working database." According to the Office Action, the preceding text clearly indicates that a separate device external to said database is an example of one of many useful features that support the rule programmer.) (Page 19, section 4.2).

As per claims 17 and 20, the Office Action indicates that Dayal teaches a method wherein said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database (i.e. "When the triggering event occurs, the condition is evaluated against the database; if the condition is satisfied, the action is executed.") (Page 2, section 1).

2. The Prior Art Reference

Dayal teaches integrating a production rules facility into a database system and providing a uniform mechanism for a number of advanced database features including integrity constraint enforcement, derived data maintenance, triggers, alerters, protection, version control, and others. In addition, Dayal teaches that a database system with rule processing capabilities provides a useful platform for large and efficient knowledge-base and expert systems.

B. The Appellants' Position

1. Independent Claims 1 and 13-15

Appellants respectfully traverse the rejections in the Office Action of independent claims 1 and 13-15 based on the following discussion. The Appellants respectfully but strongly disagrees that the claimed invention is anticipated by Dayal. More particularly, independent claims 1 and 13-15 contain features, which are patentably distinguishable from Dayal. Specifically, claims 1 and 13-15 recite, in part, "...registering alarms associated with a start and end of a lifespan of each temporal event; selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints; and upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event." These features are neither taught or suggested in Dayal because in Dayal the temporal events are not physically completely removed from the database.

Page 6 of the Office Action of October 6, 2006 suggests that Dayal teaches "registering alarms associated with a start and end of a lifespan of each temporal event" and states that the following phrase teaches this element of the Appellants' claimed invention: "a rule is triggered whenever one or more of its triggering operation occurs. In addition, active database systems must provide mechanisms for event detection and rule triggering, for condition testing, for rule action execution, and for user development of rule applications." However, there is nothing in the above selections of the Office Action's interpretation of Dayal that remotely teaches "registering alarms associated with a start and end of a lifespan of each temporal event." Nonetheless, the Office Action concludes that "an ordinary person skilled in the art *may* reasonably infer that such alarms are anticipated in the database system, because for an active database system to be active, it must do event detection, and registering alarms illustrates such a point" (emphasis added). However, the Office Action offers no tangible proof or prior art reference as to how one of ordinary skill in the art would infer this about alarms and offers no tangible evidence to support this proposition other than providing an unsubstantiated conclusionary statement unsupported by Dayal.

Furthermore, MPEP §2131 states that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The argument presented in the Office Action does not coincide with the requirements of the MPEP in that the Office Action is arguing that one skilled in the art *may reasonably infer* that the Appellants' alarms are anticipated in the database system. Rather, under a rejection under 35 USC 102(b) and under MPEP §2131 Dayal must teach every element of the claim; there is nothing in MPEP §2131 that states that a proper rejection may use a standard such as that provided in the Office Action (i.e., "may reasonably infer"). Here, Dayal fails to teach "registering alarms associated with a start and end of a lifespan of each temporal event." Furthermore, the Office Action offers no additional prior art reference that teaches this as well. While, multiple references may be used in a rejection under 35 USC 102 (see MPEP §2131.01), there are only three instances when such a use of multiple references is permissible. First, to prove the primary reference contains an enabled disclosure. Second, to explain the meaning of a term used in the primary reference. Third, to show that a characteristic not disclosed in the reference is inherent. With respect to the first instance, enablement of the Dayal reference is not at issue here. With respect to the second and third instances, the Office Action offers no such tangible reference that explains the meaning of the terms used in the Dayal reference with respect to the alarms and the Office Action offers no such tangible reference that explains that the undisclosed characteristic regarding registering the alarms associated with a start and end of a lifespan of each temporal event is inherent. Absent such a tangible reference, the rejection is deficient and improper. Additionally, merely stating that one of ordinary skill in the art would understand this does not properly constitute extrinsic evidence because the rejection lakes clarity in indicating that the missing descriptive matter is necessarily present in Dayal. The reason behind this is that the Appellants do not have the ability to read the reference that provides this information and determine whether it is proper within the context of the Appellants' invention and whether it is combinable with Dayal in the manner suggested by the Office Action. Accordingly, the rejection based on 35 USC 102(b) is improper for failing to provide tangible evidence in support of the rejection.

Page 7 of the Office Action of October 6, 2006 suggests that Dayal teaches "upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarms associated with the permanent removed temporal event" and states that the following phrase teaches this element of the Appellants' claimed invention: "Rules refer to particular tables, and so are subject to the same controls as other metadata objects (e.g. views, constraints); thus if a table is dropped, all rules defined for it are no longer operative." However, there is nothing in the above selections that remotely teaches permanent removal of the alarms from the database. Nonetheless, the Office Action concludes that an ordinary person skilled in the art may reasonably infer that the table can be removed from the database and thereby all temporal constraints and alarms associated with the targeted table can be manually removed." However, this is not what the Appellants' claimed invention teaches. The Appellants' claimed invention clearly refers to permanent removal of the alarms from the database. However, the Office Action is stating that Dayal teaches inoperable rules. Clearly, one of ordinary skill in the art would surmise that an inoperable rule is wholly distinct from a permanently removed alarm. Furthermore, the Office Action offers no tangible proof or prior art reference of how one of ordinary skill in the art would infer this about permanent removal of the alarms and offers no tangible evidence to support this proposition other than providing an unsubstantiated conclusionary statement unsupported by Dayal.

Furthermore, MPEP §2131 states that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The argument presented in the Office Action does not coincide with the requirements of the MPEP in that the Office Action is arguing that one skilled in the art *may reasonably infer* that the Appellants' permanently removed alarms are anticipated in Dayal. Rather, under a rejection under 35 USC 102(b) and under MPEP §2131 Dayal must teach every element of the claim; there is nothing in MPEP §2131 that states that a proper rejection may use a standard such as that provided in the Office Action (i.e., "may reasonably infer"). Here, Dayal fails to teach "upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarms associated with the permanent removed temporal event." Furthermore, the Office Action offers no additional prior art reference that teaches this as well. While, multiple references may be used in a rejection under 35 USC 102 (see MPEP §2131.01), there are only three instances when such a use of multiple references is permissible. First, to prove the primary reference contains an enabled disclosure. Second, to explain the meaning of a term used in the primary reference. Third, to show that a characteristic not disclosed in the reference is inherent. With respect to the first instance, enablement of the Dayal reference is not at issue here. With respect to the second and third instances, the Office Action offers no such tangible reference that explains the meaning of the terms used in the Dayal reference with respect to the alarms and the Office Action offers no such tangible reference that explains that the undisclosed characteristic regarding permanent removal of the alarms is inherent. Absent such a tangible reference, the rejection is deficient and improper. Additionally, merely stating that one of ordinary skill in the art would understand this does not properly constitute extrinsic evidence because the rejection lakes clarity in indicating that the missing descriptive matter is necessarily present in Dayal. The reason behind this is that the Appellants do not have the ability to read the reference that provides this information and determine whether it is proper within the context of the Appellants' invention and whether it is combinable with Dayal in the manner suggested by the Office Action. Accordingly, the rejection based on 35 USC 102(b) is improper for failing to provide tangible evidence in support of the rejection.

Section 2, Page 3, Paragraph 3 of Dayal merely states that:

> Like any object, rules can be created, deleted, or modified. In

> addition, rule objects have some special operations, including: **fire**, which causes a rule to be triggered; **enable**, which causes a rule to be activated; **disable**, which causes a rule to be deactivated (so that it won't be triggered even if its triggering event occurs).

The above language of Dayal makes clear that the rules that are disabled are merely deactivated, but are not permanently removed (or deleted) from the database. The Office Action offers no concrete evidence or reference in support of its conclusion that one of ordinary skill in the art would infer this to be equivalent. Indeed, such is not the case. Furthermore, Dayal indicates that the deactivated rule is merely not triggered when it becomes disabled, which indicates that the deactivated rule still remains on the database; it simply is not triggered when disabled. However, having these deactivated rules on the database system still require that the rules are residing on the database system. This is disadvantageous in that database memory is being utilized on deactivated elements. Conversely, in the Appellants' claimed invention database memory is inherently saved due to the permanent removal of the alarms.

Again, Dayal is different from the Appellants' claimed invention, which permanently removes the temporal events and corresponding alarms from the database based upon the changed temporal constraints. Hence, in the Appellants' claimed invention a deactivated rule will not be triggered because it will not exist on the database, whereas in Dayal, a deactivated rule will not trigger because it is merely disabled (but still exists on the database and consumes processing resources nonetheless). This is a significant and patentable difference between the Appellants' claimed invention and Dayal because by permanently removing the triggers from the database significantly increases the overall system efficiency, thereby improving the overall system performance including system response time (as indicated in the Appellants' FIGS. 7-9).

Furthermore, there is nothing in Dayal that suggests incorporating alarms associated with a start and end of a lifespan of each temporal event, as the Appellants' claimed invention provides. Accordingly, Dayal is missing at least one element that the Appellants' claimed invention clearly provides, which under 35 U.S.C. §102 renders the Appellants' claimed invention patentable over Dayal. Therefore, the Appellants respectfully submit that Dayal does not teach or suggest the features defined by independent claims 1 and 13-15 and as such, claims 1 and 13-15 are patentable over Dayal. In other words, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claims 1 and 13-15.

2. Dependent Claims 2-12 and 16-20

Appellants respectfully traverse the rejections in the Office Action of dependent claims 2-12 and 16-20 based on the following discussion.

(a) Dependent Claim 2

Page 7 of the Office Action of October 6, 2006 indicates that page 17, paragraph 2 of Dayal teaches the Appellants' dependent claim 2. However, the cited portion of Dayal merely refers to a database system aborting transactions containing errors, whereas the Appellants' claimed invention removes from the database temporal events that cannot evaluate as true. A reasonable reading of Dayal indicates that the error-containing transaction, while aborted, still remains on the database; it simply is not processed when aborted. However, having these aborted transactions on the database system still require that they reside on the database system. A simple analogy is when one deletes a file from his/her computer and it automatically goes to the "Recycle Bin" of the computer. However, the file still remains on the computer until the

Recycle Bin is emptied. In the current situation, and by analogy, Dayal's error-containing transaction would simply be aborted to the Recycle Bin, but would not be permanently removed from the system as Dayal offers no teaching of completely removing such transactions from its system. This is disadvantageous in that database memory is being utilized on non-processed and aborted elements. Conversely, in the Appellants' claimed invention database memory is inherently saved due to the permanent removal of the temporal events that cannot evaluate as true. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 2.

(b) Dependent Claim 3

The Office Action cites page 17, paragraph 3 of Dayal as teaching the Appellants' claimed invention of "limiting the lifespan of an event to the overlapping period of the lifespan of a parent event." However, closer scrutiny of the cited portion of Dayal reveals no such teaching. Page 17, paragraph 3 (and paragraph 2 for context) of Dayal merely states:

> The nested transaction model used in HiPAC allows some of these possibilities [terminate execution of that rule and continue rule processing, to return to the state preceding rule processing and resume database processing, or to restart rule processing]. When a rule execution subtransaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction.

There is nothing in the above-quoted language that remotely refers to the Appellants' "limiting the lifespan of an event to the overlapping period of the lifespan of a parent event." Rather, Dayal merely suggests that when a rule execution fails, then the failure event is returned to the parent node, which may repair the error. However, there is no teaching or reasonable suggestion in Dayal that indicates that the lifespan of the event is limited to the overlapping period of the lifespan of the parent event. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 3.

(c) Dependent Claim 4

The Office Action cites page 11, paragraph 1 and page 12, paragraph 1 of Dayal as teaching the Appellants' claimed invention of "changing the lifespan of an event to omit periods in which the event cannot evaluate as true." Page 11, paragraph 1 of Dayal provides:

> Another issue is whether more than one rule can be triggered by the same event; some languages preclude this, and others allow it. If it is allowed, then is the execution sequential, using some form of conflict resolution to select one rule at a time, or can rules execute concurrently? Some languages have sequential execution semantics, while others allow concurrent execution. With either sequential or concurrent execution semantics, there is also the issue of whether one rule can trigger the execution of another rule or of (another instance of) the same rule. Clearly, if such nested triggering is allowed, termination is a concern.

The above-quoted language of Dayal makes no reference to changing the duration or term of the event's lifespan, let alone changing the event's lifetime to omit periods in which the event cannot evaluate as true. Rather, this portion of Dayal merely refers to when multiple rules are being triggered by a common event, then the rules may be executed one at a time or concurrently. Also, this portion of Dayal refers to instances of using one rule to trigger another rule. The Appellants' claimed language as provided in dependent claim 4 has nothing to do with this.

Page 12, paragraph 1 of Dayal states:

> If multiple tuples are inserted into the employee table before this rule is executed, then the rule's action will retrieve all of the inserted tuples whose value in column *name* is "Bob". In general, when a triggered rule is executed in Ariel, the rule processes the entire set of triggering changes, including both the user-generated changes that initiated rule processing and any subsequent changes made by rule actions. If a rule is executed multiple times during rule processing (e.g., because it is re-triggered by another rule's changes, or because it triggers itself), then each time it executes, it processes all matching changes since the last time it executed. If **rollback** is executed in a rule action, then rule processing terminates and the transaction is aborted.

The Office Action states that an ordinary person skilled in the art would understand that termination or abort must take place when the event cannot evaluate as true. First, there is nothing in the above-language of Dayal that refers (either explicitly or implicitly) that termination or abort must take place when the event cannot evaluate as true. Moreover, the Office Action offers no tangible proof or prior art reference as to how one of ordinary skill in the art would infer this and offers no tangible evidence to support this proposition other than providing an unsubstantiated conclusionary statement unsupported by Dayal.

Furthermore, MPEP §2131 states that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). The argument presented in the Office Action does not coincide with the requirements of the MPEP in that the Office Action is arguing that one skilled in the art *understands* that the Appellants' lifespan of an event are changed to omit periods in which the event cannot evaluate as true. Rather, under a rejection under 35 USC 102(b) and under MPEP §2131 Dayal must teach every element of the claim; there is nothing in MPEP §2131 that states that a proper rejection may use a standard such as that provided in the Office Action (i.e., conclusionary statement of what one of ordinary skill in the art "understands").

Here, Dayal fails to teach "changing the lifespan of an event to omit periods in which the event cannot evaluate as true." Furthermore, the Office Action offers no additional prior art reference that teaches this as well. While, multiple references may be used in a rejection under 35 USC 102 (see MPEP §2131.01), there are only three instances when such a use of multiple references is permissible. First, to prove the primary reference contains an enabled disclosure. Second, to explain the meaning of a term used in the primary reference. Third, to show that a characteristic not disclosed in the reference is inherent. With respect to the first instance, enablement of the Dayal reference is not at issue here. With respect to the second and third instances, the Office Action offers no such tangible reference that explains that how the multiple execution of the rules in Dayal or the circumstances of when a rollback is executed in a rule action in Dayal would be understood by one skilled in the art to teach "changing the lifespan of an event to omit periods in which the event cannot evaluate as true" or that such features are inherent. Absent such a tangible reference, the rejection is deficient and improper. Additionally, merely stating that one of ordinary skill in the art would understand this does not properly constitute extrinsic evidence because the rejection lakes clarity in indicating that the missing descriptive matter is necessarily present in Dayal. The reason behind this is that the Appellants do not have the ability to read the reference that provides this information and determine whether it is proper within the context of the Appellants' invention and whether it is combinable with Dayal in the manner suggested by the Office Action. Accordingly, the rejection based on 35 USC 102(b) is improper for failing to provide tangible evidence in support of the rejection. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 4.

(d) Dependent Claim 5

The Office Action cites page 3, paragraph 5 and page 17, paragraph 3 of Dayal as teaching the Appellants' claimed invention as defined by dependent claim 5. The cited language of Dayal provides:

> In addition, some languages provide mechanisms whereby data (parameters) can be *bound* in the event and/or condition part of a rule, then passed to the condition and/or action.
>
> The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution subtransaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction.

There is nothing in the above-quoted language that refers to linking (assigning) the lifespan of an event that does not have a defined lifespan as the lifespan of a parent event. Rather, the above-quoted language simply refers to binding data in the event and/or condition portion of a rule, and when a rule fails to execute, having the failure event return to the parent, which may either issue a new transaction or repair the error. There is nothing in the cited portion of Dayal regarding using the lifespan of the parent event for an event having no defined lifespan. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 5.

(e) Dependent Claim 6

The Office Action cites page 17, paragraph 3 of Dayal as teaching the Appellants' claim

6. However, this portion of Dayal is bereft of any teaching of the Appellants' "propagating the lifespan or context of the parent node to all children nodes of the parent node." Rather, this portion of Dayal merely provides:

> The nested transaction model used in HiPAC allows some of these possibilities. When a rule execution subtransaction fails, the failure event is returned to its parent, which has the option of spawning a sibling subtransaction to repair the error (this may be accomplished through the firing of another rule that is triggered by the failure event). Alternatively, failure can be propagated up the transaction tree all the way to the root (top) transaction.

Clearly, the above-quoted language simply refers to the situation when a rule fails to execute, having the failure event return to the parent, which may either issue a new transaction or repair the error. There is nothing regarding using the lifespan or context of the parent node to all children nodes. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 6.

(f) Dependent Claim 7

The Appellants' claim 7 is dependent on independent claim 1, and as such, all of the limitations of claim 1 should be incorporated into claim 7 for purposes of patentability. As previously described, Dayal fails to teach all of the elements of the Appellants' independent claim 1; *ipso facto* Dayal must fail to teach all of the elements of Appellants' claim 7 by dependency. In fact, the mechanism in Dayal of establishing a lifetime of an event is by the use of a rule. However, the Appellants' claims are not so limiting. It should be noted that the Office Action cites, in part, page 5, paragraph 2 of Dayal as teaching "[s]ome languages support rules triggered by temporal events. These might be absolute (e.g., 08:00:00 hours on 1 January 1994), relative (e.g., 5 secs. After takeoff), or periodic (e.g., 17:00:00 hours every Friday)." Actually, page 5, paragraph 2 of Dayal provides:

> The SQL2 standard allows *assertions* to be defined on tables. Each assertion is a simple rule that is triggered by one of the following events: **before commit, after insert, after delete,** or **after update** of a table. In the case of updates, a subset of the table's columns may be specified, so that the rule is triggered only when those columns are updated. The proposed SQL3 standard introduces **triggers** in addition to assertions (the difference will become clear when we discuss the action parts of these rules). The allowed triggering events are **before** or **after** an insertion, deletion, or update of a table.

Accordingly, page 5, paragraph 2 of Dayal does not provide any of the language that the Office Action purports as teaching the Appellants' claim 7. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 7.

(g) Dependent Claim 8

The Appellants' claim 8 is dependent on dependent claim 4, which is dependent on independent claim 1, and as such, all of the limitations of claims 1 and 4 should be incorporated into claim 8 for purposes of patentability. As previously described, Dayal fails to teach all of the elements of the Appellants' independent claim 1 and dependent claim 4; *ipso facto* Dayal must fail to teach all of the elements of Appellants' claim 8 by dependency. In fact, the mechanism in Dayal of establishing a lifetime of an event is by the use of a rule. However, the Appellants' claims are not so limiting. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 8.

(h) Dependent Claim 9

The Appellants' claim 9 is dependent on dependent claim 4, which is dependent on independent claim 1, and as such, all of the limitations of claims 1 and 4 should be incorporated into claim 8 for purposes of patentability. As previously described, Dayal fails to teach all of the elements of the Appellants' independent claim 1 and dependent claim 4; *ipso facto* Dayal must fail to teach all of the elements of Appellants' claim 9 by dependency. In fact, the mechanism in Dayal of establishing a lifetime of an event is by the use of a rule. However, the Appellants' claims are not so limiting. It should be noted that the Office Action cites, in part, page 5, paragraph 2 of Dayal as teaching "[s]ome languages support rules triggered by temporal events. These might be absolute (e.g., 08:00:00 hours on 1 January 1994), relative (e.g., 5 secs. After takeoff), or periodic (e.g., 17:00:00 hours every Friday)." Actually, page 5, paragraph 2 of Dayal provides:

> The SQL2 standard allows *assertions* to be defined on tables. Each assertion is a simple rule that is triggered by one of the following events: **before commit, after insert, after delete,** or **after update** of a table. In the case of updates, a subset of the table's columns may be specified, so that the rule is triggered only when those columns are updated. The proposed SQL3 standard introduces **triggers** in addition to assertions (the difference will become clear when we discuss the action parts of these rules). The allowed triggering events are **before** or **after** an insertion, deletion, or update of a table.

Accordingly, page 5, paragraph 2 of Dayal does not provide any of the language that the Office Action purports as teaching the Appellants' claim 9. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 9.

(i) Dependent Claims 10 and 18

The Office Action cites page 5, paragraph 2 and page 6, paragraph 3 of Dayal as teaching the Appellants' "combining events using a sequence operator to form a composite event having a time span." It should be noted that the Office Action cites, in part, page 5, paragraph 2 of Dayal as teaching "[s]ome languages support rules triggered by temporal events. These might be absolute (e.g., 08:00:00 hours on 1 January 1994), relative (e.g., 5 secs. After takeoff), or periodic (e.g., 17:00:00 hours every Friday)." Actually, page 5, paragraph 2 of Dayal provides:

> The SQL2 standard allows *assertions* to be defined on tables. Each assertion is a simple rule that is triggered by one of the following events: **before commit, after insert, after delete,** or **after update** of a table. In the case of updates, a subset of the table's columns may be specified, so that the rule is triggered only when those columns are updated. The proposed SQL3 standard introduces **triggers** in addition to assertions (the difference will become clear when we discuss the action parts of these rules). The allowed triggering events are **before** or **after** an insertion, deletion, or update of a table.

Accordingly, page 5, paragraph 2 of Dayal does not provide any of the language that the Office Action purports as teaching the Appellants' claims 10 and 18. Furthermore, the Office Action cites, in part, page 6, paragraph 3 of Dayal as teaching "[w]hen an instance of this event type occurs, the formal parameters are bound to a specific employee (the one whose salary is being updated) and two specific integers (this employee's old salary and new salary)." Actually, page 6, paragraph 3 of Dayal provides:

> In SQL2 assertions, the condition also is a predicate, but the condition is satisfied if the predicate is false. In HiPAC, rule conditions are sets of predicates or queries on the database; if all of the predicates are satisfied and all of the queries' results are non-empty, then the condition is satisfied. Transition conditions may be expressed in HiPAC using the event parameter mechanism described in Section 2.1.

Accordingly, page 6, paragraph 3 of Dayal does not provide any of the language that the Office Action purports as teaching the Appellants' claims 10 and 18. Rather, page 4, last paragraph and page 5, paragraph 4 provide, "[s]ome languages support rules triggered by temporal events. These might be absolute (e.g., 08:00:00 hours on 1 January 1994), relative (e.g., 5 secs. After takeoff), or periodic (e.g., 17:00:00 hours every Friday)" and "[w]hen an instance of this event type occurs, the formal parameters are bound to a specific employee (the one whose salary is being updated) and two specific integers (this employee's old salary and new salary)" respectively. However, there is no teaching in Dayal in these sections or anywhere else that indicates that the time span taught in page 4, last paragraph of Dayal may be combined with the formal parameters and integers in page 5, paragraph 4 of Dayal. In fact, the "instance of this event type" indicated in page 5, paragraph 4 of Dayal refers to HiPAC events (object-oriented events), while the events referred to in page 4, last paragraph of Dayal refer to temporal events (which may be different from object-oriented events). Accordingly, the events discussed in these two separate sections of Dayal are different and unique and there is no indication in Dayal that such events may be interchanged and the characteristics of one may be imported into the characteristics of the other or that they may be combined in the manner suggested in the Office Action. Additionally, there is nothing in the cited portions of Dayal that refer to using a "sequence operator to form a composite event" as provided by the Appellants' claimed language. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claims 10 and 18.

(j) Dependent Claim 11

The Office Action cites a portion of page 10, paragraph 3 of Dayal as teaching the Appellants' claimed invention as defined by dependent claim 11, "associating a lifespan with the sequence operator." The relevant portion of page 10, paragraph 3 provides:

> Most importantly, unlike in AI systems, in active database systems rule processing is integrated with conventional database activity-queries, modifications, and transactions-and it is this activity that causes rules to become triggered and initiates rule processing.

The Office Action concludes that the above-quoted text "*clearly* indicates that a lifespan is contained within a rule that processes queries, modifications, and transactions, and the sequence operator is the active database system that initiates rule proceesing" (emphasis added). However, the Appellants' challenge this conclusion on the grounds that the above-quoted language does not refer to a lifespan of an event or associating the lifespan of an event with a sequence operator, and that one skilled in the art would hardly read into the above-quoted language what the Office Action purports as being clear. In fact, such a reading is overly broad and unduly burdonsome with respect to one skilled in the art, who, at the time of the invention, would not have the luxury of the hindsight reasoning provided by the Office Action. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 11.

(k) Dependent Claim 12

The Appellants' claim 12 is dependent on independent claim 1, and as such, all of the limitations of claim 1 should be incorporated into claim 12 for purposes of patentability. As previously described, Dayal fails to teach all of the elements of the Appellants' independent claim 1; *ipso facto* Dayal must fail to teach all of the elements of Appellants' claim 12 by dependency. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claim 12.

(l) Dependent Claims 16 and 19

The Office Action cites page 19, section 4.2 of Dayal as teaching the Appellants' "using a separate device external to said database to detect the combined events." However, a closer review of the cited portion of Dayal reveals no such teaching. Pages 19-20, section 4.2 of Dayal provides:

> The implementation of an active database system can include many useful features that support the rule programmer. Features for analyzing rule processing include the ability to trace rule execution, to display the current set of triggred rules, to query and browse the set of rules, and to cross-reference rules and data. Other useful features include the ability to control errors in rule programs, to activate and deactivate selected rules or groups of rules while the databse system is processing transactions, and to experiment with rules on an off-line subset of a working database. Simple versions of some of these features exist in some active database systems, while more sophisticated and complete versions will certainly emerge over time.

The Office Action concludes that the above-quoted language of Dayal "*clearly* indicates that a suparate device external to said database is an example of one of many useful features that support the rule programmer" (emphasis added). However, such an indication is neither *clear* nor supported by the language in Dayal. There is nothing in the above-quoted language in Dayal that specifically addresses using a separate device that is external to the database to detect the combined events. Rather, Dayal merely makes reference to experimenting with rules on an off-line subset of a working database. However, an "off-line subset of a working database" does not infer that (1) the subset is a device that (2) is separate from the database that is (3) external to the database and that is (4) used for detecting the combined events. The Appellants' claims 16 and 19 provide for all four features. However, Dayal addresses none of them, yet inexplicably, the Office Action concludes that Dayal makes this teaching *clear*. One skilled in the art would hardly read into the above-quoted language what the Office Action purports as being clear. In fact, such a reading is overly broad and unduly burdonsome with respect to one skilled in the art, who, at the time of the invention, would not have the luxury of the hindsight reasoning provided by the Office Action. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claims 16 and 19.

(m) Dependent Claims 17 and 20

The Office Action cites page 2, [paragraph] 1 as teaching the Appellants' claimed invention as provided by dependent claims 17 and 20, "wherein said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database." Page 2, paragraph 1 of Dayal provides:

> This allows rules to be triggered by events such as database operations, by occurrences of database states, and by transitions between states (among other things), instead of being evaluated by an inference engine that cycles periodically trhough the rules. When the triggering event occurs, the condition is evaluated against the database; if the condition is satisfied, the action is executed. Rules are defined and stored in the database, and evaluated by the database system, subject to authorization, concurrency control, and recovery.

As provided in the above-quoted language of Dayal, there is no teaching of restricting the event (i.e., "consists") to an "instantaneous and atomic point of occurrence within an application." Furthermore, the above-quoted language of Dayal merely indicates that an action is executed upon the evaluation of condition against the database with respect to a triggered event. However, there is no indicating in Dayal that the instantaneous and atomic point of occurrence of the event affects the state of the database. Rather, the database in Dayal is merely affected by the executed action and not by an event's instantaneous and atomic point of occurrence. Accordingly, Dayal fails to teach each and every feature of the Appellants' claimed invention as required under 35 U.S.C. §102(b). Therefore, the Board is respectfully requested to reconsider and withdraw the rejections to claims 17 and 20.

C. Conclusion

In view of the foregoing, the Appellants respectfully submit that the cited prior art, Dayal, does not teach or suggest the features defined by independent claims 1 and 13-15, and as such, claims 1 and 13-15 are patentable over Dayal. Further, dependent claims 2-12 and 16-20 are similarly patentable over Dayal, not only by virtue of their dependency from patentable independent claims, respectively, but also by virtue of the additional features of the Appellants' claimed invention they define. Thus, the Appellants respectfully request that the Board reconsider and withdraw the rejections of claims 1-20 and pass these claims to issue.

Please charge any deficiencies and credit any overpayments to Attorney's Deposit Account Number 09-0441.

Respectfully submitted,

Date: February 28, 2007

/Mohammad S. Rahman/
Mohammad S. Rahman, Esq.
Registration No. 43,029

Gibb I.P. Law Firm, LLC
2568-A Riva Road, Suite 304
Annapolis, MD, 21401
Voice: (301) 261-8625
Fax: (301) 261-8825
Customer No. 29154

IX. CLAIMS APPENDIX

1. A method of monitoring events in a database, said method comprising:

storing in said database at least one database rule;

mapping temporal constraints of an event of the database rule to corresponding temporal events;

changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule;

registering alarms associated with a start and end of a lifespan of each temporal event;

selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints; and

upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event.

2. The method as claimed in claim 1, further comprising removing from the database temporal events that cannot evaluate as true.

3. The method as claimed in claim 1, further comprising limiting the lifespan of an event to the overlapping period of the lifespan of a parent event.

4. The method as claimed in claim 1, further comprising changing the lifespan of an event to omit periods in which the event cannot evaluate as true.

5. The method as claimed in claim 1, further comprising assigning a lifespan of an event having an undefined lifespan as the lifespan of a parent event.

6. The method as claimed in claim 1, further comprising propagating the lifespan or context of the parent node to all children nodes of the parent node.

7. The method as claimed in claim 1, wherein a lifespan of an event is expressed as a predetermined duration of time.

8. The method as claimed in claim 4, wherein the lifespan is dependent upon the associated event.

9. The method as claimed in claim 4, wherein the lifespan ends at a predetermined time, or recurs at a predetermined period of time.

10. The method as claimed in claim 1, further comprising combining events using a sequence operator to form a composite event having a time span.

11. The method as claimed in claim 7, further comprising associating a lifespan with the sequence operator.

12. The method as claimed in claim 1, further comprising storing a database rule as an event-condition-action (ECA) rule.

13. A database recorded on a computer storage medium comprising:

software code means for storing in said database at least one database rule;

software code means for mapping temporal constraints of an event of the database rule to corresponding temporal events;

software code means for changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule;

software code means for registering alarms associated with a start and end of a lifespan of each temporal event;

software code means for selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints; and

software code means for, upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event.

14. A system for monitoring events in a database, said system comprising:

means for storing in said database at least one database rule;

means for mapping temporal constraints of an event of the database rule to corresponding temporal events;

means for changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule;

means for registering alarms associated with a start and end of a lifespan of each temporal event;

means for selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints; and

means for, upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event.

15. A program storage device readable by computer, tangibly embodying a program of instructions executable by said computer to perform a method of monitoring events in a database, said method comprising:

storing in said database at least one database rule;

mapping temporal constraints of an event of the database rule to corresponding temporal events;

changing said temporal constraints associated with the temporal events based upon temporal constraints for related events of said database rule;

registering alarms associated with a start and end of a lifespan of each temporal event;

selectively deploying and selectively permanently removing the temporal events from said database based upon the changed temporal constraints; and

upon reaching said end of said lifespan of said each temporal event, permanently removing from said database said alarm associated with the permanently removed temporal event.

16. The method of claim 10, further comprising using a separate device external to said database to detect the combined events.

17. The method of claim 1, wherein said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database.

18. The program storage device as claimed in claim 15, wherein said method further comprises combining events using a sequence operator to form a composite event having a time span.

19. The program storage device of claim 18, wherein said method further comprises using a separate device external to said database to detect the combined events.

20. The program storage device of claim 15, wherein said event consists of an instantaneous and atomic point of occurrence within an application that affects the state of said database.

X. EVIDENCE APPENDIX

There is no other evidence known to Appellants, Appellants' legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no other related proceedings known to Appellants, Appellants' legal representative or Assignee which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.